universität innsbruck

Summer Term 2019


# Interactive Theorem Proving using Isabelle/HOL

## Session 13

Christian Sternagel

Department of Computer Science

## Topics

calculational reasoning, case analysis, code generation, computation induction, data type invariants, document preparation, finding theorems, first steps, functional programming in HOL, higher-order logic, history and motivation, induction, inductive definitions, Isabelle basics, Isabelle/Isar, Isabelle/ML, IsaFoR/CeTA, locales, manual termination proofs, multisets, natural deduction, notation, proof methods, PSL: a high-level proof strategy language, rule induction, rule inversion, session management, sets, simplification, sledgehammer, structural induction, structured proof, The Archive of Formal Proofs, the certification approach, total recursive functions, type classes, type definitions, well-foundedness

# Overview

- The Archive of Formal Proofs
- The Certification Approach
- The IsaFoR/CeTA Project
- Exercises

## The Archive of Formal Proofs

- collection of Isabelle/HOL formalizations
- available from https://www.isa-afp.org/
- maintained by Isabelle developers (that is, kept up to date with latest Isabelle version)
- submit your own development through web-interface

## Using the AFP

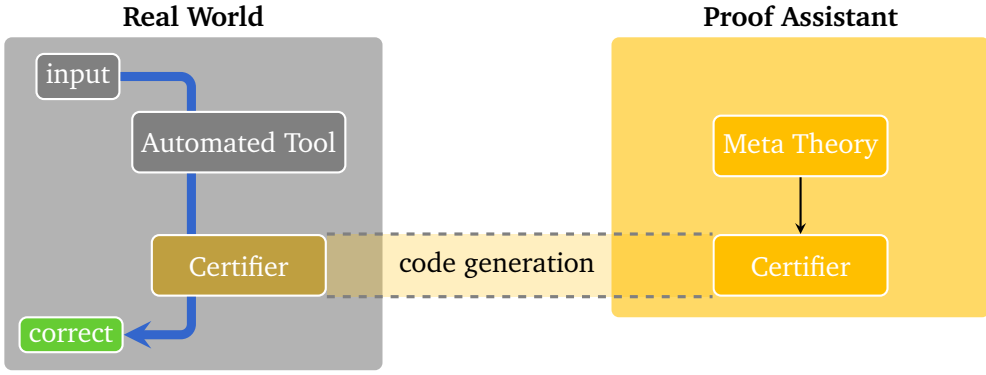
(1) download the AFP tar-archive and extract it into `$AFP`

(2)
```
$ echo "$AFP/thys" >> ~/.isabelle/Isabelle2018/ROOTS
```

## Demo13.thy – First-Order Terms

- apply substitution $\sigma = \{y \mapsto \mathsf{g}(\mathsf{b})\}$ to term $t = \mathsf{f}(x, y)$: $t\sigma = \mathsf{f}(x, \mathsf{g}(\mathsf{b}))$
- unify terms $\mathsf{f}(\mathsf{a}, x)$ and $\mathsf{f}(x, \mathsf{b})$: ✘
- unify terms $\mathsf{f}(\mathsf{a}, x)$ and $\mathsf{f}(y, \mathsf{b})$: ✔ with mgu $\{x \mapsto \mathsf{b}, y \mapsto \mathsf{a}\}$

# The Certification Approach




- implement untrusted automated tool
- formalize meta-theory of tool in proof assistant
- use formalized meta-theory to verify certifier for tool output
- certifier bridges real world and logical world of proof assistant

## IsaFoR/CeTA

- available from http://cl-informatik.uibk.ac.at/isafor
- specific instance of certification approach for term rewriting

## Example – Termination of Odd Even/Odd Function

- ```
  fun evenodd :: "nat ⇒ bool ⇒ bool"
    where
      "evenodd 0 True ⟷ True"
    | "evenodd x False ⟷ ¬ (evenodd x True)"
    | "evenodd (Suc x) True ⟷ evenodd x False"
  ```
- a corresponding term rewrite system (TRS)
- Tyrolean Termination Tool 2 (ttt2) proves termination automatically
- verified certifier CeTA validates resulting proof
  ```

# Exercises

work on your semester projects